HUMAN-MACHINE INTERFACE – SEARCHING FOR THE OPTIMAL SUPPORT IN THE COCKPIT

Jiří Kacer, Jiří Fischer

University of Defence, Brno, Czech Republic

jiri.kacer@unob.cz, jiri.fischer@unob.cz

Abstract: *Paper deals with a part of the Adaptive cockpit systems project – System situation recognizer module, which was included in the simulation cockpit of the F-16 fighter. In the text are shown the fundaments of the project, the ideas why and how can adaptive cockpit help by using and utilization of the aircraft abilities in the air combat. Then will be present, how was the model of the System situation recognizer module created and the reader will be informed about the off-line tests, how they were prepared, how they were made and what were the results of the simulation.*

1 DAPTIVE COCKPIT SYSTEM PROJECT BACKGROUND

The Adaptive Cockpit System (ACS) project is oriented to support the pilot in the air combat situation, when he is overloaded from the enemy and from all systems on the board of the aircraft. The aim of the project is to improve the pilot – aircraft interface using the cybernetics principles, advanced information technology and results of progress in aviation and in other allied branches of science and technology. That led to formulation of NLR's project ACS - Adaptable Cockpit System. The consideration, resources and objectives are formulated in Memorandum VE-2000-002 and it has been elaborated between NLR (Nationaal Lucht- en Ruimtevaartlaboratorium, The Netherlands) and MAB (Military academy in Brno, Czech Republic) in 2003.

The role of the pilot in the future air combat is expected to remain essential. However, technological advances are required to provide a "competitive edge" and involve increasingly sophisticated tools for executing the mission. The challenge will be to create an environment that:

- allows humans to deploy their unique qualities adequately, and
- fully exploit the power of the technical resources available to support the mission.

Advances in information assessment, artificial intelligence, psycho-physiological processing, and human-interface technology create opportunities for an efficient sharing or exchange of human and machine functions. This human-machine "partnership" will relieve pilots of the burden of managing tasks that are not directly or momentary not essential for mission accomplishment while allowing them to keep or acquire full awareness of the essential task and mission information context(s).

The new generation of aircraft will be equipped with *adaptive cockpit interfaces.* This kind of human-interface technology will dynamically adapt or be reconfigured in-flight according to a number of options:

- flight status, aircraft parameters and mission phase,

- actual performance and momentary work(over)load of the individual pilot, his/her psycho-physiological state, and control mode,
- training and experience level(s) of the pilot(s).

The "adapted interface" will modify the presentation of data, depending on the state and capacity of the pilot. It only presents information the pilot really needs for performing and survival and will present it in the most effective mode and/or location. The general idea behind this interactive technology is to ensure that the pilot is able to guide and use his attention in the most optimal way to prevent overload. Hence, an optimal `supportive' *adaptable cockpit system* (ACS) that acts as a so-called `team player' is provided. The following main information and data elements should be `researched' to enable dynamic adaptation and reconfiguration in the future:

- dynamic assessment of the goals and sub-goals of a flight mission because information about the mission should be assessed on line (aircraft mission phase data and tactical situation),
- real-time monitoring and determination of the pilot's state and behavior (control mode, intent, possible errors, workload levels) and estimated human processing capabilities (on line experience level, fatigue, etc.),
- interpretation of the pilot's state data and the use of rules of thumb to determine the most likely opportunities for adaptation.

Earlier research has concentrated on assessing (deriving) vehicle-state in the mission but with varying levels of success, mainly due to ineffective means of communicating between man and machines. The goal of the present study is the definition and evaluation of a prototype adaptable interface technique to identify the specific automation requirements and practical utility of this innovation in military cockpits.

2 SYSTEM SITUATION RECOGNISER

System situation recogniser (SSR) is the basic subsystem of ACS. It is presented as a system, that is operating with the data of the pilot and the actual technical state of the system on one side and on the other side is the environment around. The outcomes are the values of the situation set.

The essential part of the system is its base of knowledge, which describes a part of reality relevant to the system. The base of knowledge is declarative module of SSR, expresses general relationship between the input data measured and the value of the situation set. The system is understood as the pilot, the machine and the environment, where are these located. The knowledge is primarily describing possible situations. That means comprehensive view of the momentary situation. Knowledge domain is trying to describe the system, which is not globally known and to write it as a casual relation. Therefore it is necessary to choose the right admittance, which is considered of characterization of the system behaviour, where is identification of typical sequences of system change used and represented as procedures.

Situation is more general than state. It considers actual physical condition of the pilot and his actual behaviour, technical state of the machine, direct and indirect outer influence, environmental condition and state of the mission target. The change of the situation can be understood as sequence of characteristically changes – procedures. Than the behaviour of the system can be indicated that way, that each procedure will be taken as one entity. It can be described and the relationships can be described as well.

The desiderative content of situation set is briefly said the condition of the pilot, the plane and its systems, the environment around, the action they're taking and the mission goal. The operation is understood in the tactical meaning (in what phase of the mission is the system located) as well as the technical aspect (what are the technical subsystems doing).

Based on the required structure of situation set is the knowledge hierarchically ordered into 4 layers (see Fig. 1), which are sorted in the context "mission". The first and highest layer is "tasks". Task is essential time and function part in the mission, which is describing single parts of the mission, mainly the tactical point of view. In the context "mission" are single tasks always disjunctive (there are never two or more tasks running at same time) and simultaneously during the mission can't be a situation, which no task is running.

The next layer is "subtasks". Subtask is standardized and important procedure in range of the Task. The difference is that it doesn't have only tactical character, but the technical site is included as well. Subtask doesn't have to be disjunctive, but it is possible, that more than one subtask at time is running. It is also possible, that no subtask is active. Subtask can be included in more than one task, and in each of them can have different parameters (the characteristics have to be the same).

The third layer is "function". Function is an entity telling us the state or behaviour of some part of the system based on given attributes. Two types of functions are used.

The state function (diagnostic) has information about the state (e.g. biological condition of the pilot or the technical situation of the plane) in the logical or ordinal values.

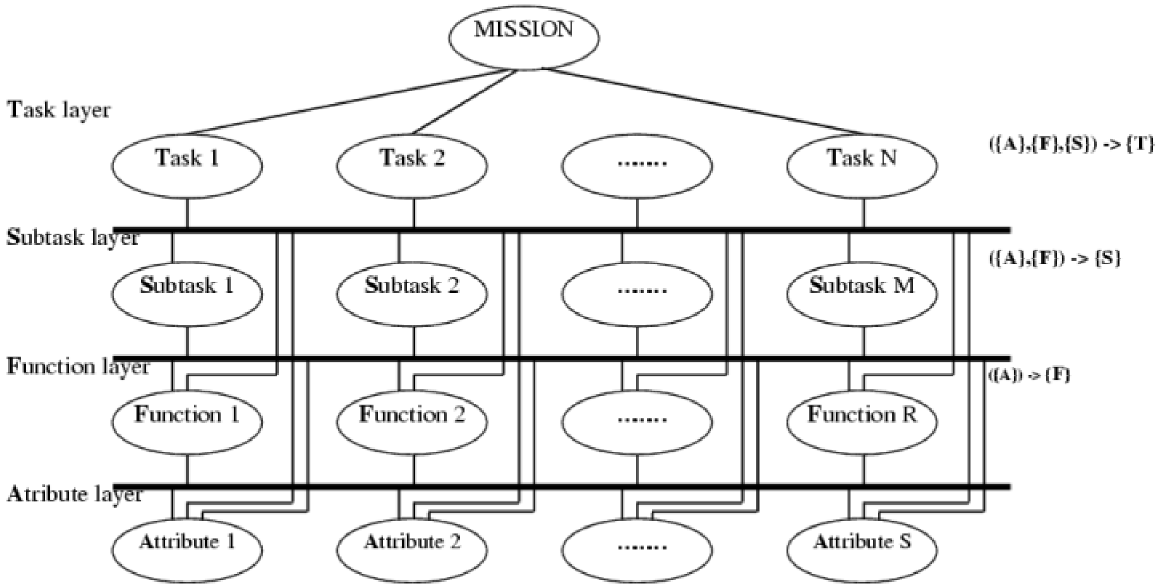

Fig. 1: The 4-layer structure Knowledge domain SSR

Procedural function testifies the "in motion progression" of procedural changes. (e.g. the prior to the start of the plane is under way). The function, that describes situation from the technical site, it is also possible (and expected) concurrency of more than one function.

On the lowest layer are attributes. These are entities determined based on the data of the real situation, or based on the results of former identification calculations. Attributes are generally taking on values in the discrete range {0,1, …, 100}, or the $\perp$ value, which expresses the undefined value.

Single entities standing for sequence of changes of the situation (procedures) is appropriate to describe using typical patterns. These are the first way of representation of the knowledge domain. The patterns are the input for the process, that in the preliminary (off-line) phase on its base makes a group of predicate rules and a group of identifiable rules, that are qualified for computer processing and are used for inference (in the on-line phase).

On the Fig. 2 is conducted the analysis diagram of the SSR knowledge domain. The knowledge is firstly described and registered all over and step-by-step decomposed to single entities.

2.1 Description of attributes and entities

The knowledge base is including word description of attributes and their semantics. Next the description of functions, subtasks and tasks. Verbal description of the knowledge is important for system handling, which is making the patterns of single procedures and defines relevant attributes and their semantics. Semantics of attributes is understood verbal listing of possible values, that the attribute can gain and its description.

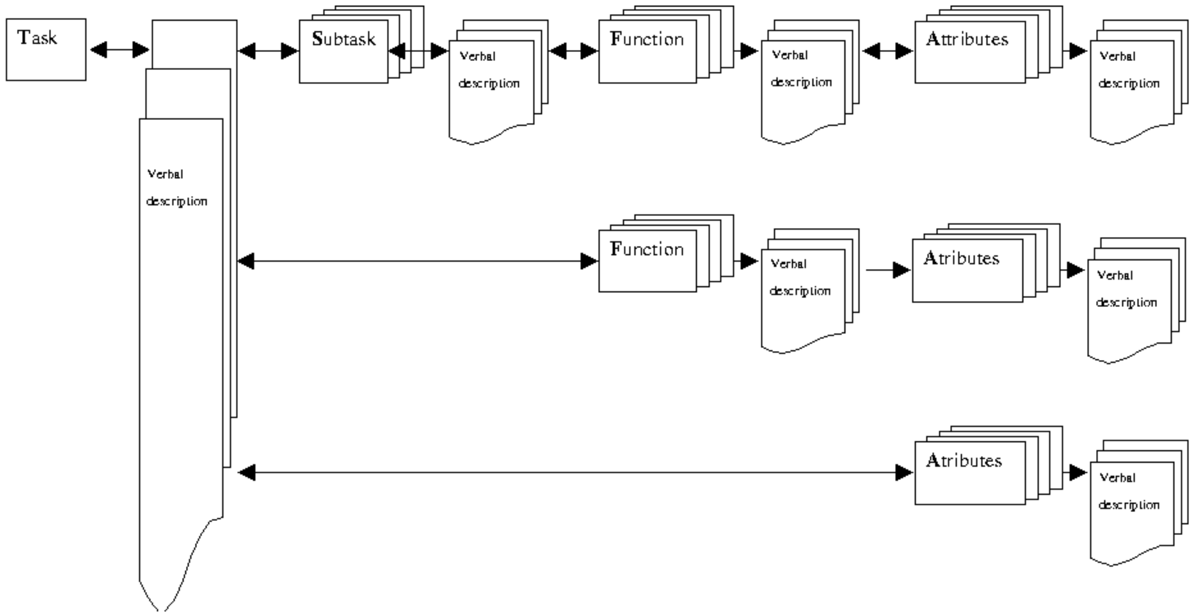

Fig. 2: The SSR knowledge domain analysis – definition of tasks, subtasks and functions

Attribute is a quantity, that is holding information deduced from the measured data and it is in a form (language), that is suitable for application in LHS production rules of SSR. The language of representation is based on an assumption line of variability of the attribute, which is given by the subset of whole numbers $A \in 0, 1, 2 \ldots, n$; that enables in one language to represent:

Ordinal quantities, which have their relation order preserved. In case the attribute is representation a quantity, that assumes values in extent of real numbers (or generally in extent, that is bigger, than the extend of attribute values), we will set up fitting interval resolution. For example "height h [hm]" is represented like this:

$h = 0$	$A = 0$
$h \in (0, 1>$	$A = 1$
$h \in (k - 1, k>$	$A = 2$

Logical quantities are taking only 0 and 1 values on. Generally by the attributes, which describe state, that can be expressed by YES / NO, for example switched on (A = 1) / switched off (A=0).

Linguistics quantities, where just one or none possibility comes. Extend is subset of extend of attributes, that are carrying linear value. Relation of order doesn't make sense. For example quantity "mode 1 - 3" is represented by:

Mode 1 A = 1

Mode 2 A = 2

Mode 3 A = 3

Attributes are further arranged to several groups due to the source of the value:

Attributes – the value is determined based on the quantities measuring (output State assessor module and Pilot assessor module), they're tagged Av.

Attributes – the value is determined based on attributes Av and a set of currently running functions, they're tagged Af.

Attributes – the value is determined based on attributes Av, Af and a set of currently running subtasks, they're tagged AS.

Entities describe single procedures on all of the structure levels of knowing, that means functions, subtasks and tasks. Described is a usual entity characteristic that is for example usual procedure duration time and its variation. The description of the entities is closely banded with the patterns, because the entity de-facto characterizes the procedure succession that is including.

2.2 Identification rules

During the mission the interference module is working with the knowledge represented by the set of identification rules. These rules are describing the present situation and running procedures based on values measured and normalized quantities. The SSR is using rules for on-line operation, by evaluation and identification of the present situation.

We understand F as a set of functions, S as a set of subtasks, T as a set of tasks and Av, Af and AS relations above the attributes. Each function is characterized by the values of input attributes Av, subtask by the values of the attributes Af and the task by the values of the attributes AS.

The set of identification rules I is possible to describe as display $I = I_f \cup I_s \cup I_t$, where

$I_f : A_v \to F$

$I_s : A_f \to S \times \omega$

$I_t : A_s \to T \times \omega$

ω is designating weight of the rule (credibility that the procedure is running)

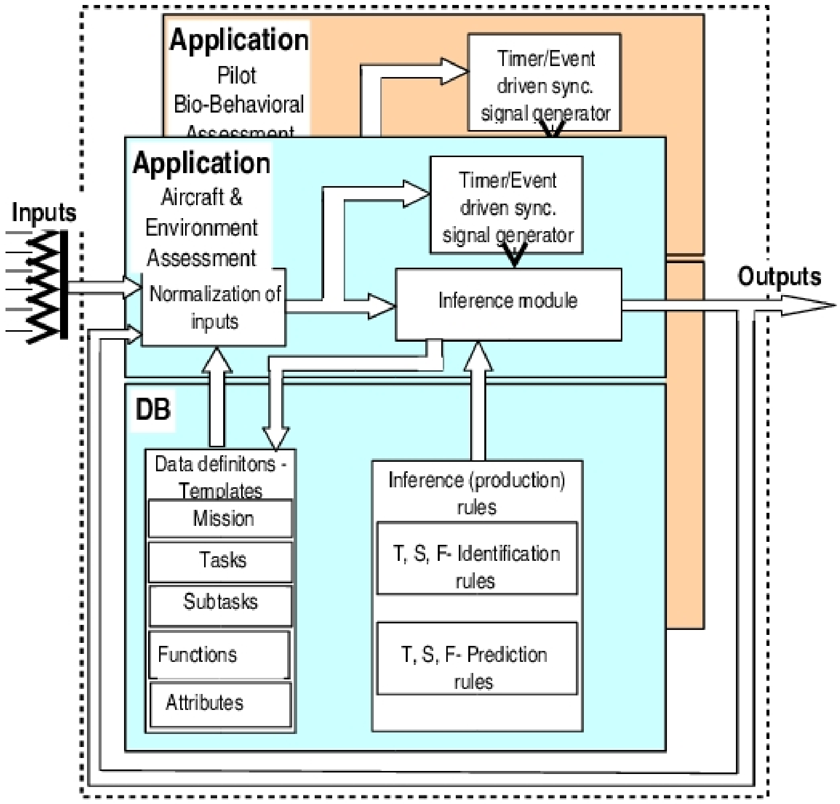

Fig. 3. The SSR implemental structure

3 IMPLEMENTATION OF THE SSR

One of the goals of the project is also to make functional referential implementation. The scheme of that model according to analysis of the system is shown in Fig. 3. The base claims are simplicity and lucidity. On the other side the effectiveness of implementation, hardware needs, used platform and massiveness of the system is not seen as that important. This approach is in the world of software evolution is a little bit unusual, but it is important to realize these facts: The main purpose of this referential implementation is to test the system as one unit, the testing and support of knowledge base evolution. Important task is also to find and eliminate conception defects and problems earlier, than the real implementation is done. On the other side will not happen that the system in this appearance will be tested directly in the planes.

Due to disintegration of single parts of ACS project among several solvers I am doing mainly SSR subsystem implementation that is forwarded in Brno. The interface between SSR and other subsystems of ACS is still in the solving phase. The supporting parts of SSR are identification mechanism and knowledge base (see the Fig. 4). To these is mounted a set of tools appointed to prepare data for the knowledge base and results of the output.

3.1 Used program software

Identification mechanism is solved in the system C++. Interference is presently implemented as the main process, that apart of its own evaluation providing needed inputs and outputs – mainly for communication with the knowledge base, normalization and processing of inputs from the environment and formulation of output to related modules of the system (reasoning module).

The knowledge base contains several more or less independent parts, but for software is the implementation united. The main quest for this base was the capability of quick answer on a specific set of questions like “find me the set of entities, that have its value of attribute $A_1 = 1$”, because the questions of this type occur most frequently during the interference process. He led to that, that the novice was searched among present implementations of relation database systems. After deciding about other circumstances and based on agreement among solving teams was the relation database system MySQL voted. It was because of several things:

- it is dispensed under GNU / GPL license

- many software and hardware platforms support it – MS Windows, Linux, various UNIX versions…

- many tools to convert to other frequent relation database systems are available

- it is quite efficient with small and medium large systems

- in many various software setting is supporting MySQL

- many supporting tools are available (e.g. systems to administrate the database using a web browser)

It was created new module called prosaically SSR-server that is offering the identifiable mechanism a set of functions for certain questions to the knowledge base. This toolbox is consecutively working with MySQL database, that will process the input to the fitting format, will do the needed operations with the database and the results will process to the appearance that is suitable again for SSR toolbox its self.

3.2 Structure of the implementation

Rough structure of the system in present phase of evolution from the software equipment is shown on the Fig. 4. Calculation process is running in the service module that is written C++ system. It is doing all the calculating stuff during active phase of SSR

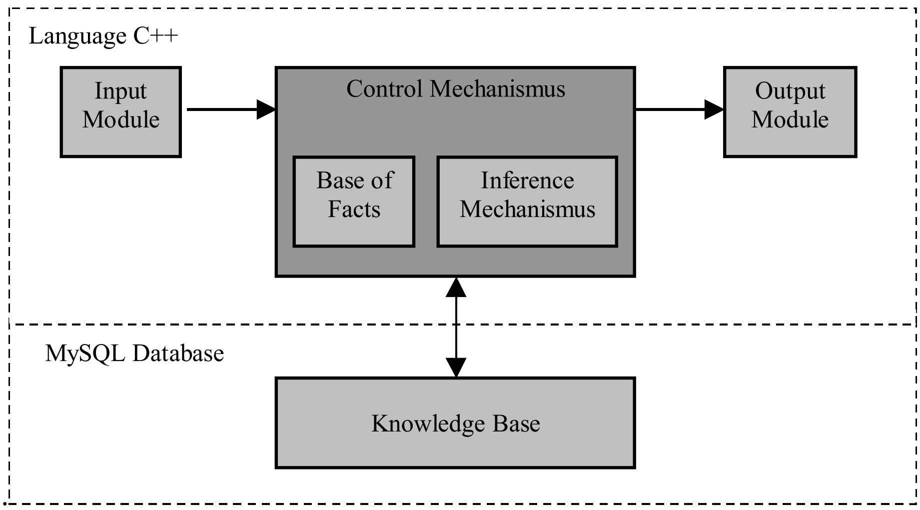

Fig. 4. The SSR implemental structure

That means normalization and valuation of the input data (pilot assessment, situation assessment); identification of the situation, prediction based to present situation and saving the present state to the history kept in the database. This procedure is repeated in never-ending loop during all the active phase of SSR.

The knowledge base is (as the previous chapters announce) represented in relation DB of My SQL system, as a set of simple tables with simple relations.

3.3 Off-line tests

Because the aim of the ACS project is to support the pilot of the fighter F-16 in difficult situations, it is necessary to test the whole system. In the first phase of the testing of the right system function are made the off-line tests. The advantage of these tests is to finalize the running of prepared software in "own home" environment without the need to run expensive flight simulator and without pilots. These tests were made upon data acquired from any earlier simulation flight mission performed on the flight simulator. All data are recorded in a text file and the values of the measured valuables are ordered in the columns. They are then converted as attributes and used in the form of questions to the MySQL Database to find the recent functions, subtasks a task. The described process is shown on the Fig 5.

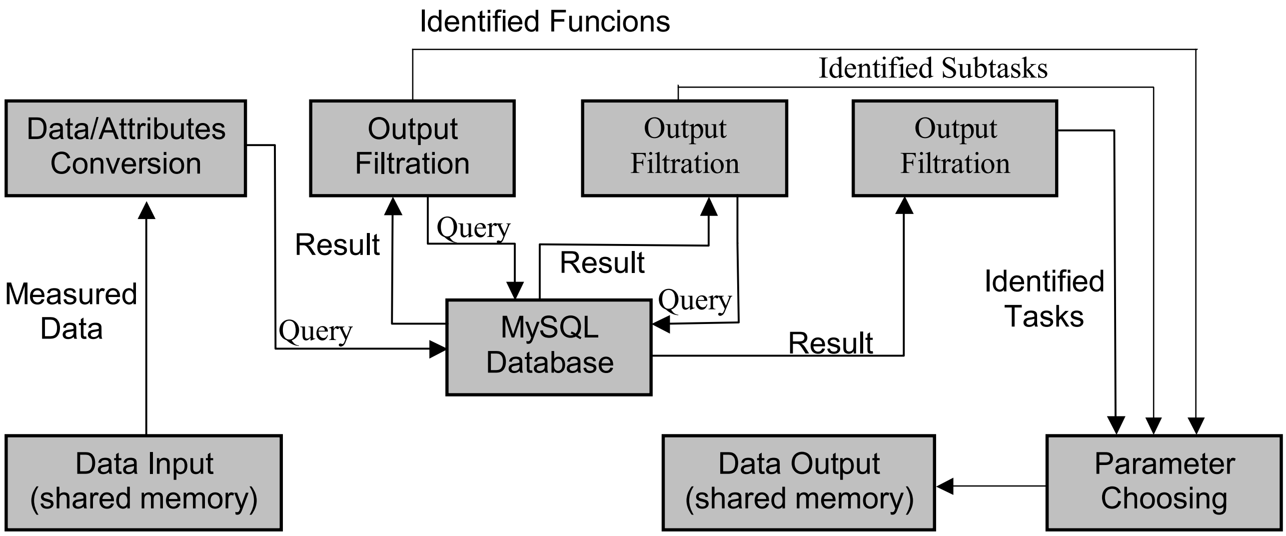

Fig. 5. The SSR-server

Their values are recorded in other text file and selected parameters are written in the shared memory, where are they prepared for next processing in the reasoning module of the ACS.

4 CONCLUSIONS

The desiderative content of situation set is briefly said with the condition of the pilot, the plane and its systems, the environment around, the action they're taking and the mission goal. The operation is understood in the tactical meaning (in what phase of the mission is the system located) as well as the technical aspect (what are the technical subsystems doing). Based on the required structure of situation set is the knowledge hierarchically ordered into 4 layers, which are sorted in the context "mission".

During the mission the inference module is working with the knowledge represented by the set of identification rules. These rules are describing the present situation and running procedures based on values measured and normalized quantities. The SSR is using rules for on-line operation, by evaluation and identification of the present situation.

One of the goals of the project is also to make functional referential implementation. The main purpose of referential implementation is to test the system as one unit, the testing and support of knowledge base evolution. Important task is also to find and eliminate conception defects and problems earlier, than the real implementation is done. On the other side will not happen, that the system in this appearance will be tested directly in the planes.

Identification mechanism is solved in C++. Interference is presently implemented as the main process, that apart of its own evaluation providing needed inputs and outputs – mainly for communication with the knowledge base, normalization and processing of inputs from the environment and formulation of output to related modules of the system (reasoning module).

The SSR server is making interface between the service module and the knowledge base on the level of C++. The main type of question on the knowledge base is identification questions that mean questions finding entities based on the values (some) of attributes.

The aim of the off-line tests – verifying the right function of the SSR module - was fulfilled and the created software could detect the flight phase. The next step in the testing is the on-line test, which will be made at NLR with help of the pilots flying predefined mission on the flight simulator.

5 REFERENCES

Bohnen, H.G.M, Reus, A.J.C., Vollebregt, A.M. & Beetstra, J.W. (2003). Adaptive Cockpit Systems: System functionality and architecture, National Aerospace Laboratory (NLR), NLR-CR-2002-559.

Řeřucha, V., Kacer, J., Řeřucha, Š., Janíček, J.,: The SSR – The System Situation Recogniser (SSR), WP4 – Final Report, NLR, The ACS Project, Amsterdam, The Netherlands / Military Academy in Brno, Czech Republic. June 2004. 24 pp.